Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

205

Paul Klint

A Study in String Processing Languages

Springer-Verlag
Berlin Heidelberg New York Tokyo

Author

Paul Klint
Centre for Mathematics and Computer Science
Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

CR Subject Classification (1985): D.3.1, D.3.3, F.3.3, I.7.0

ISBN 3-540-16041-8 Springer-Verlag Berlin Heidelberg New York Tokyo
ISBN 0-387-16041-8 Springer-Verlag New York Heidelberg Berlin Tokyo

Printing and binding: Beltz Offsetdruck, Hemsbach/Bergstr.
2145/3140-543210

CONTENTS

PART II: Definition of SUMMER

PREFACE

Written text is an essential element in our culture and various technical means have been invented to aid in its production. Paper and pencil, the typewriter and the typesetter are examples of such inventions.

Continuing this same line of development, computers are nowadays being used to alleviate the writing task. Computerized text processing systems (ranging from word processors for writing and editing simple texts to fully automated newspaper and book printing systems) are rapidly penetrating into all areas of human activity where written text is the primary means of communication.

Historically, the impetus behind the development of computers has always been primarily numerical in nature. This is reflected in the design of most computers and programming languages. However, the increasing use of computers for text processing and other non-numeric tasks makes the purely arithmetic design obsolete.

This monograph concentrates on the programming language aspects of computerized text handling and, to be more precise, on the design and implementation of **string processing languages**. The term 'string processing' refers to the process of inspecting, modifying and transforming texts, i.e. sequences of symbols. It comprises such seemingly disparate activities as text editing, transforming a text with embedded formatting directives into a final layout, and compiling a source program into a string of machine instructions.

An account is given of attempts to solve some of the problems in string processing languages. First of all, an exercise in designing an application oriented programming language is described. This has resulted in the language SUMMER. Next, an exercise in the formal definition of the semantics of programming languages is described. The formal definition and implementation of SUMMER together constitute the final result of the project.

This monograph consists of two parts. Part I is devoted to string processing languages in general and consists of chapters 1 through 5. Part II is devoted to the definition of SUMMER and consists of chapters 6 through 10. The contents of the monograph are now briefly summarized.

Chapter 1 is introductory and gives the necessary motivation and background for the study of string processing languages.

Chapter 2 is devoted to general design considerations for string processing languages and compares the semantics of various pattern matching models. Attention is paid to different forms of side-effects during a pattern match. This is done by giving an operational, formal definition of the semantics of the various models. As a result of this, a new pattern matching model based on side-effect recovery is developed.

Chapter 3 gives an overview of the language SUMMER. SUMMER may be characterized as a **small** language, i.e. it consists of a relatively small set of primitive operations together with a modest extension mechanism.

Chapter 4 concentrates on the problem of finding a method for formal language definition that is suitable for the designers as well as the implementors and users of a language. An improved method for the operational definition of programming language semantics is developed and the result of applying this method to SUMMER is illustrated.

Chapter 5 concludes the first part of this monograph with an evaluation of the research described in it and suggestions for further research.

Part II is devoted to the definition of the SUMMER programming language. It provides both a formal and informal language definition and tutorial examples.

In Chapter 6 the techniques and notational conventions that are used in the definition are introduced. Much attention is paid to the method used for the formal definition of the semantics of SUMMER.

Chapter 7 contains a semi-formal definition of the SUMMER kernel. This is a small subset of the language on which a semantic definition of the whole language can be based. The description of each language feature consists of its syntax, an informal as well as a formal definition of its semantics, and examples.

In Chapter 8 the kernel is extended with useful data types and associated operations, such as reals, arrays, tables, files, bit strings, etc. Some complete, annotated SUMMER programs are presented in Chapter 9. Finally, a summary of the syntax is given in Chapter 10.

Readers who are only interested in getting a general impression of the language SUMMER may confine themselves to Chapter 3 and the annotated examples in Chapter 9. Readers who are not interested in the formal definition of the language may skip Chapter 6 (except Sections 6.1 and 6.2), and all subsections of Chapter 7 entitled 'Semantics'.

This monograph is a revision of the author's dissertation as presented at the Technical University Eindhoven, The Netherlands. It was supervised by Prof. dr. F.E.J. Kruseman Aretz (Technical University Eindhoven, The Netherlands) and Prof. H. Whitfield, b.s., d.i.c. (University of Newcastle upon Tyne, Great Britain). The research was conducted while the author was employed at the Mathematical Centre, Amsterdam, The Netherlands.

Several people contributed to this effort. Design and implementation of SUMMER (as well as of its predecessor SPRING) were done in close cooperation with Marleen Sint and contributions to the design of SUMMER were made by Jan Heering. Their enthusiasm, patience and friendship were essential to the success of this project.

Jan Heering, Marleen Sint and Arthur Veen have read drafts of this monograph. They pointed out various errors and made numerous suggestions for improving the style and presentation of it. I am grateful for their support and criticism. Comments made by Leo Geurts, R.J. Lunbeck, Lambert Meertens and W.L. van der Poel are gratefully acknowledged.